Proceedings, 1st International Conference on Image Processing (ICIP-94), Austin, Texas, November 1994.

OPTIMAL DATA PARTITIONING OF MPEG-2 CODED VIDEO

Alexandros Eleftheriadis and Dimitris Anastassiou

Department of Electrical Engineering
and Center for Telecommunications Research
Columbia University, New York, NY 10027, USA
{eleft,anastas}@ctr.columbia.edu

ABSTRACT

We analyze the problem of optimal data partitioning of MPEG-2 coded video in an operational rate-distortion context. The optimal algorithm is characterized and shown to have high complexity and delay. A causally optimal algorithm based on Lagrangian optimization is proposed, that optimally solves the problem for intra (I) pictures, while it provides an optimal solution for predicted/interpolated (P/B) pictures when the additional constrains of causal operation and/or low-delay are imposed. A memoryless version of the algorithm, theoretically optimal for intra-pictures only, is shown to perform almost identically but with significantly less computational complexity. Finally, a fast, suboptimal algorithm using slice-based optimization is also proposed, and is shown to perform quite close (within 1 dB) to the causally optimal one.

1. INTRODUCTION

Data partitioning is a feature of the MPEG-2 draft standard which provides for the segmentation of a coded signal bitstream into two components or partitions [1]. It can be a very effective tool for the transmission of video over channels that allow selective protection of each of the partitions. Channels of this type can be implemented, for example, using increased forward error correction, or employing high priority transmission in an ATM-based (Asynchronous Transfer Mode) networking environment. By transmitting the most critical information with high reliability, i.e. over the highest quality channel, the average quality of the signal reconstructed at the receiver can be significantly increased for the same level of channel distortion. This feature is one of the major benefits of pyramidal or–more generally–hierarchical, multi-layered coding schemes.

An important characteristic of data partitioning is that it can be employed even after encoding has taken place, in contrast with other hierarchical approaches, such as the SNR, spatial, or temporal scalability modes of MPEG-2. This is because the encoder does not need to maintain a prediction loop per each signal layer, a necessary requirement for a pyramidal scheme in which each coding layer is an enhancement of its previous one. As a direct consequence, it is also less robust in the sense that neither partition is self-contained; loss of information in either one will cause error propagation and accumulation during the decoding process if temporal predictive/interpolative modes are used. Although data partitioning is currently supported only in MPEG-2, it is trivial (at least syntactically) to incorporate into other coding schemes.

We provide an analysis of optimal data partitioning in a rate-distortion sense, for the partitioning of MPEG-2 compliant bitstreams. Although no assumptions need to be made in the analysis, a constant source bit rate is assumed and used for simplicity in our derivations and simulations; the extensions to the general case are relatively straightforward. The optimal algorithm is characterized, and shown to have significantly high complexity and delay. A *causally optimal* algorithm based on Lagrangian optimization is described; it optimally solves the problem for intra (I) pictures, while it provides an optimal solution for predicted/interpolated (P/B) pictures when the additional constrains of causal operation and/or low-delay are imposed. A memoryless version of the algorithm, theoretically optimal for intra-pictures only, is shown to perform almost identically but with significantly less computational complexity. Finally, a fast, suboptimal algorithm using *slice-based* optimization is also proposed, and is shown to perform quite close (within 1 dB) to the causally optimal one.

2. OPTIMAL DATA PARTITIONING

The system diagram of the data partitioning scheme is shown in Figure 1. In between an MPEG-2 encoder/decoder pair, the bitstream (assumed here to be coded at the constant rate of B Mbps) is split into two parts, each being transmitted on a different "channel". Throughout this paper we will assume that channel 0 is a perfect one (no losses, errors, or insertions) with a given fixed available bandwidth $\hat{B} < B$, while channel 1 is assumed to exhibit stochastic behavior.

Partitioning is performed at well-defined points (breakpoints) in the bitstream syntax. For our purposes, and to ensure that partition 0 is independently decodable, we will constrain the allowable breakpoint values so that critical quantities such as macroblock address increment and DCT DC differential values (for intra-coded macroblocks) are included in partition 0[1]. Consequently, partitioning will only affect the number of coefficient run-length codes that will

[1] Widely adopted MPEG terminology is used throughout this paper; good descriptions are given in [1, 4].

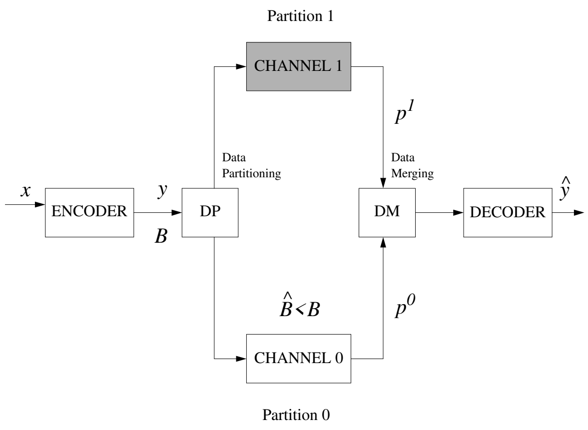

Figure 1: Data partitioning system

be carried in partition 0, while the rest will be assigned to partition 1. Note that the breakpoint value is the same for all blocks of a given slice. Sequence headers are replicated in partition 1 to increase robustness, and hence the total rate for the transmission of the signal is slightly increased.

Denoting y the coded video signal, $\hat{y}$ the output of the decoder, p^i the signal of the i-th partition, and $R(\cdot)$ the rate, the problem of optimal data partitioning takes the form:

$$\min_{R(p^0)\leq\hat{B}} \{\|y-\hat{y}\|\} \tag{1}$$

Since channel 1 is assumed to exhibit stochastic behavior, we consider the deterministic problem of minimizing the maximum possible error, i.e.:

$$\min_{R(p^0)\leq\hat{B}} \{\max\{\|y-\hat{y}\|\}\} \tag{2}$$

This corresponds to the case where the entire partition 1 is lost.

The optimization window in (2) is not specified, and it can span from just a part of a picture, up to any number of pictures. In general, and taking into account that data partitioning as described here is performed after encoding has taken place, it is desirable to keep the end-to-end delay low. Computation complexity considerations impose additional constrains on the window length. Consequently, we will typically be interested in solutions of (2) that consider up to a single complete picture.

An important aspect of the problem not readily evident in (2) is its recursive nature, caused by the corresponding recursive process with which y and $\hat{y}$ are generated (decoded) when P and B pictures are involved. In the following we separately consider two cases: intra-picture only partitioning, and mixed-mode (I, P, and B) partitioning.

3. INTRA-PICTURE PARTITIONING

In intra-picture only partitioning, there is no temporal dependence between pictures. Consequently, the partitioning error will simply consist of the DCT coefficients that were assigned to partition 1. Using the orthonormality of the DCT, this can be expressed as follows:

$$\min_{R(p^0)\leq\hat{B}} \{\max\{\|y-\hat{y}\|\}\} \Longleftrightarrow$$

$$\min_{\sum_{i=1}^{N} R_i(B_i)\leq\hat{B}} \left\{\sum_{i=1}^{N} D_i(B_i)\right\} \tag{3}$$

with

$$D_i(B_i) \equiv \sum_{j\in S_i}\sum_{k\geq B_i} [E_j^i(k)]^2 \tag{4}$$

and where $B_i \in \{0,\ldots,64\}$ is the breakpoint value for slice i (run-length codes from B_i and up will go to partition 1), N is the number of slices considered, S_i are the blocks in slice i, $E_j^i(k)$ is the value of the DCT coefficient of the k-th run in the j-th block of the i-th slice, and $R_i(B_i)$ denotes the rate of using breakpoint value B_i in the i-th slice in partition 0.

This optimization problem can be solved using an iterative bisection algorithm based on Lagrangian optimization, which at each step k separately minimizes $D_i(B_i) + \lambda_k R_i(B_i)$ for each slice. A similar algorithmic approach but in a different context has been used in [2, 3, 5]. The collection of necessary data in eq. 3 requires only parsing of the bitstream up to inverse quantization of the DCT coefficients (this represents a small fraction of the complete decoding process). The window (N) in which the algorithm operates is a design parameter. Since data partitioning–as was mentioned before–is performed on top of encoding, it is desirable to minimize the additional delay introduced by the extra processing step. A plausible selection is then a single picture (frame or field). The target bit budget R_{budget} of each picture can be set to: $R_{\text{budget}} = (\hat{B}/B)R - R_o$, where R is the size (in bits) of the currently processed frame, and R_o are the number of bits spend for coding components of the bitstream that are not subject to data partitioning. R is immediately available after the complete picture has been parsed. Allocated bits that are left over from one picture are carried over to the subsequent picture.

A short description of the complete algorithm is as follows. We denote by $R_i^*(\lambda)$ and $D_i^*(\lambda)$ the optimal rate and distortion respectively for slice i for that particular λ (i.e. they minimize $D_i + \lambda R_i$). We also denote by $B_i^*(\lambda)$ the breakpoint value that achieves this optimum.

Lagrangian Optimization Algorithm

Step 1: Initialization
Set $\lambda_l = 0$ and $\lambda_u = \infty$. If the inequality:

$$\sum_{i=1}^{N} R_i^*(\lambda_u) \leq R_{\text{budget}} \leq \sum_{i=1}^{N} R_i^*(\lambda_l) \tag{5}$$

holds as an equality for either side, an exact solution has been found. If the above does not hold at all, then the problem is infeasible (this can happen if the target rate $\hat{B}$ is too small). Otherwise go to Step 2.

Step 2: Bisection and Pruning

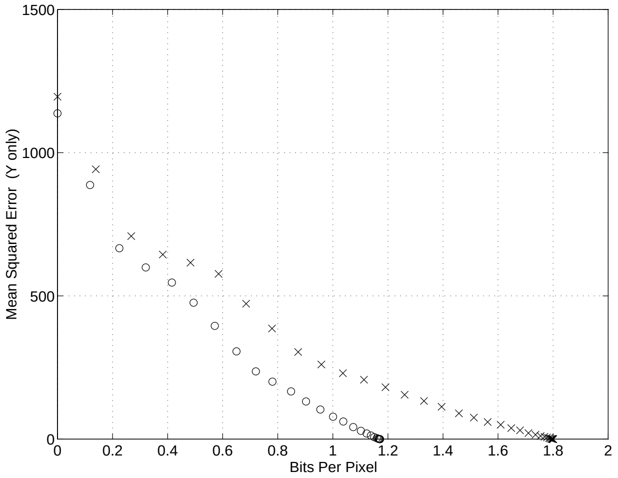

Figure 2: Slice 20 (full-width, frame 0) from "Flower Garden" coded at 24 (x) and 12 (o) Mbps.

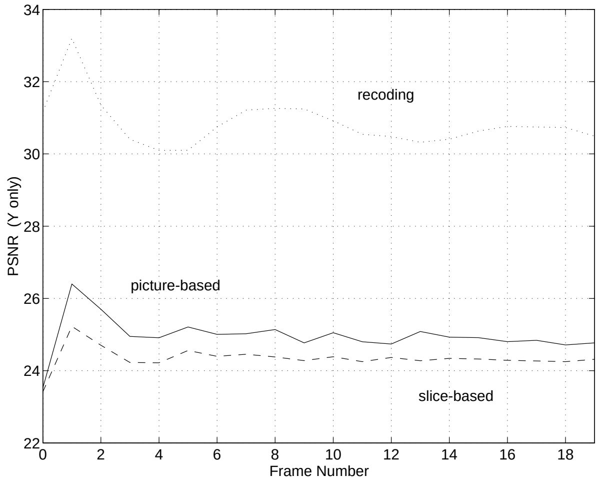

Figure 3: Data partitioning of frame-based, intra coded "Flower Garden", from 24 Mbps to 12 Mbps, using picture-based (Lagrangian) and slice-based algorithms (dotted line is for full recoding at 12 Mbps).

Compute:

$$\lambda_{\text{next}} := \left| \frac{\sum_{i=1}^{N} [D_i^*(\lambda_u) - D_i^*(\lambda_l)]}{\sum_{i=1}^{N} [R_i^*(\lambda_u) - R_i^*(\lambda_l)]} \right| \tag{6}$$

and find $R_i^*(\lambda_{\text{next}})$ and $D_i^*(\lambda_{\text{next}})$ such that $B_i^*(\lambda_u) \leq B_i^*(\lambda_{\text{next}}) \leq B_i^*(\lambda_l)$.

Step 3: Convergence Test
If

$$\sum_{i=1}^{N} R_i^*(\lambda_{\text{next}}) = \sum_{i=1}^{N} R_i^*(\lambda_u) \text{ or } \sum_{i=1}^{N} R_i^*(\lambda_{\text{next}}) = \sum_{i=1}^{N} R_i^*(\lambda_l) \tag{7}$$

then stop; the solution is $B_i^*(\lambda_u)$, $i = 1, \ldots, N$. If

$$\sum_{i=1}^{N} R_i^*(\lambda_{\text{next}}) > R_{\text{budget}} \tag{8}$$

then $\lambda_l := \lambda_{\text{next}}$, else $\lambda_u := \lambda_{\text{next}}$.

The bisection algorithm operates on the convex hull of the $R(D)$ curve of each slice. Consequently, points which lie above that, and hence are not $R(D)$ optimal, are not considered by the algorithm. Figure 2 shows the $R(D)$ plots of typical slices (frame-based, intra coding of "Flower Garden" at 24 and 12 Mbps, slice 20 — full-width — of frame 0). Worth noting is the locally non-convex behavior in both cases. This property can be traced back to the structure of the MPEG-2 run-length encoding tables, where specific examples of non-convexity can be easily found. In some cases, if the $R(D)$ curve of a slice is sufficiently misbehaved, the bisection algorithm can be set off track, with a resulting underutilization of the target bit budget. In order to mitigate this effect, and also to speed up operation, each iteration considers a continuously shrinking interval of possible breakpoint values ("pruning"). This will result in convergence of the algorithm to a much smaller set of non-convex points.

The computational overhead of the algorithm is small, and convergence is achieved within 8–10 iterations. Figure 3 shows the results (Y-PSNR, $\|y - p^0\|$) of applying the algorithm to 20 frames of "Flower Garden", using frame-based intra coding at 24 Mbps, and with a target rate of 12 Mbps for partition 0. Also shown are the results of a simpler algorithm that uses *slice-based* optimization. In this latter case each slice is independently assigned a target number of bits proportional to its original size, and a breakpoint is selected so that this budget is not exceeded (leftover bits are carried over to the next slice). Note that the slice-based algorithm is purely rate-based, i.e. the distortion is completely ignored. Lagrangian optimization outperforms the slice-based algorithm by 0.6 dB on the average. Finally, for comparison purposes, we also show the Y-PSNR of full recoding from 24 to 12 Mbps.

4. MIXED-MODE PARTITIONING

When all types of picture coding types are used (I, P, and B) the problem is significantly more complex. The decoding process can be described by:

$$P_i = \mathcal{M}_i(P_{i-1}) + e_i \tag{9}$$

where P_i denotes the i-th decoded picture (in coding order), $\mathcal{M}_i(\cdot)$ denotes the motion compensation operator for picture i, and e_i denotes the coded prediction error. The first picture is assumed to be intra-coded, and hence $P_0 = e_0$. Although, for simplicity, a single reference picture is shown above for motion compensation, the expression can be trivially extended to cover the general case (which includes B-pictures).

By applying data partitioning and decoding partition 0, equation 9 becomes:

$$\hat{P}_i = \mathcal{M}_i(\hat{P}_{i-1}) + \hat{e}_i \tag{10}$$

where $\hat{e}_i$ denotes the partitioned prediction error. Using (9) and (10), eq. (2) becomes:

$$\min_{\sum_{i=1}^{N} R_i(B_i)\le \hat{B}} \left\| \sum_{p=1}^{M} \mathcal{M}_i(P_{i-1}) - \mathcal{M}_i(\hat{P}_{i-1}) + e_i - \hat{e}_i \right\| \tag{11}$$

where M is the number of pictures over which optimization takes place. Note that in general $\mathcal{M}_i(P_{i-1}) - \mathcal{M}_i(\hat{P}_{i-1}) \neq \mathcal{M}_i(P_{i-1} - \hat{P}_{i-1})$, i.e. motion compensation is a non-linear operation, because it involves integer arithmetic with truncation away from zero.

From eq. (11) we observe that, in contrast with the intra-only case, optimization involves the accumulated error $a_i \equiv \mathcal{M}_i(P_{i-1}) - \mathcal{M}_i(\hat{P}_{i-1})$. Furthermore, due to the error accumulation process, partitioning decisions made for a given picture will have an effect in the quality and partitioning decisions of subsequent pictures. As a result, an optimal algorithm for (11) would have to examine a complete group of pictures (I-to-I), since breakpoint decisions at the initial I-picture may affect even the last B or P picture. Not only the computational overhead would be extremely high, but the delay would be unacceptable as well. It is desirable then to seek fast solutions with small delay, that are able to control error propagation in a well-defined fashion.

An attractive alternative algorithm is one that solves eq. (11) on a picture basis, and where only the error accumulated from past pictures is taken into account; this algorithm will be referred to as *causally optimal*. Note that in order to accurately compute a_i, two prediction loops have to be maintained (one for a decoder that receives the complete signal, and one for a decoder that receives only partition 0). This is because of the nonlinearity of the integer arithmetic of motion compensation. With the penalty of some lack in arithmetic accuracy, these two loops can be collapsed together.

The causally optimal problem can be formulated as follows:

$$\min_{R(p^0)\le \hat{B}} \{\max\{\|a_i + e_i - \hat{e}_i\|\}\} \iff \min_{\sum_{i=1}^{N} R_i(B_i)\le \hat{B}} \left\{ \sum_{i=1}^{N} \hat{D}_i(B_i) \right\} \tag{12}$$

with $\hat{D}_i(B_i)$ defined by:

$$\hat{D}_i(B_i) \equiv \sum_{j\in S_i} \left\{ \sum_{k} A_j^i(k)^2 + \sum_{k\ge B_i} 2A_j^i(\mathcal{I}(k))E_j^i(k) + E_j^i(k)^2 \right\} \tag{13}$$

and where N is such that a complete picture is covered, $A_j^i(k)$ is the k-th DCT coefficient (in zig-zag scan order) of the of the j-th block of the i-th slice of the accumulated error a_i, and $\mathcal{I}(\cdot)$ maps run/length positions from the prediction error $E_j^i(\cdot)$ to actual zig-zag scan positions.

The minimization problem in (13) can be solved using the Lagrangian optimization approach of Section 3, assuming that the convexity of the $R(D)$ curves is generally maintained. Figure 4 shows the R(D) curve for slice 20 of frame 3 (P-picture) from the sequence "Mobile" coded at at 4 Mbps (frame-based coding) and partitioned at 3.2 Mbps using the causally optimal algorithm. The upper curve takes into account the accumulated error a_i, whereas the bottom one involves only the prediction error partitioning distortion $e_i - \hat{e}_i$. We observe that convexity is clearly present. The property holds even for small slice sizes (e.g. 11 or 4 macroblocks per slice, instead of the regular 44 which amounts for the whole picture width), although the curves become progressively flatter.

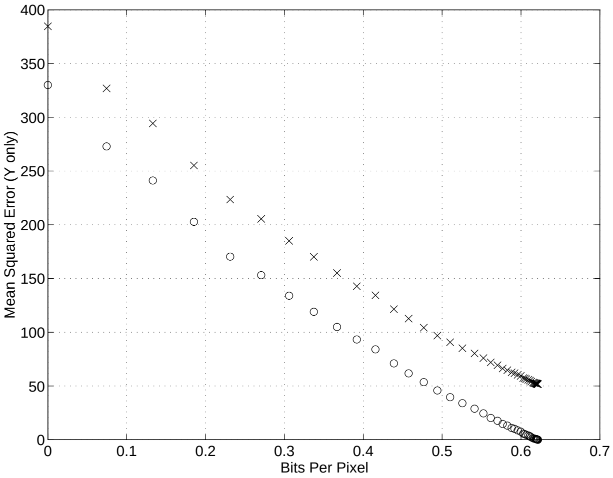

Figure 4: Slice 20 (full-width, frame 3, P-picture) from "Mobile" coded at 4 Mbps and partitioned at 3.2 Mbps: (x) $\hat{D}(B_i)$, (o) $D(B_i)$.

An important issue in mixed-mode coding is the target bit budget that will be set for each picture. In a typical situation, I and P picture DCT coding requires a significant number of bits, while B picture sizes are dominated by header and motion vector coding bits. Consequently, B pictures provide much less flexibility for data partitioning. In order to accommodate this behavior, I and P pictures are assigned proportional bit budgets as in Section 3; for B pictures the same is done, except when the resulting bit budget is negative, in which case it is set to 0. The negative budget, however, is accounted for, so that the bits spent for the B picture are subtracted from the budget of the immediately following picture. Note that an optimal bit allocation for each picture would be a direct by-product of the optimal (non-causal) algorithm.

Figure 5 shows the Y-PSNR resulting from the causally optimal algorithm on 15 frames of the "Mobile" sequence (I distance N=15, I/P frame distance M=3), frame-based coded at 4 Mbps and partitioned at 3.2 Mbps (80% of the rate goes to partition 0). This is the signal quality that would be observed by a decoder that receives only partition 0, compared with one that receives both partitions. We see that I and P frames suffer the most, while B frames are in general up to 1 dB better.

The complexity is solving equation 12 is significant, as it requires a complete decoding loop for the luminance signal. In addition, since motion compensation is performed in the spatial domain while partitioning is performed in the DCT domain, a forward DCT computation module is required as well in order to compute $A_j^i(\cdot)$. As a result, the imple-

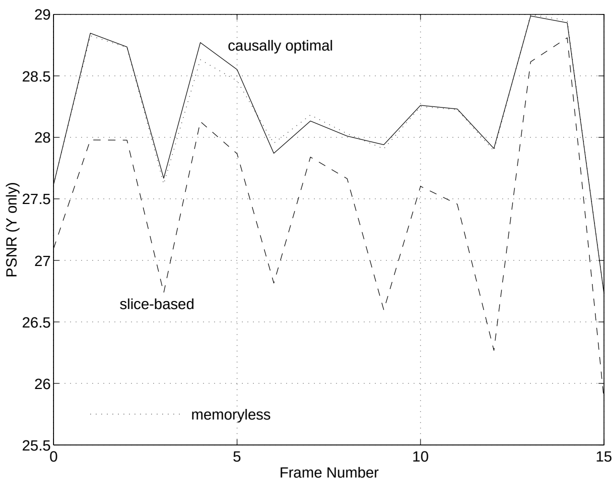

Figure 5: PSNR (Y only) for "Mobile" sequence, frame-based coded at 4 Mbps and partitioned at 3.2 Mbps using causally optimal, memoryless, and slice-based algorithms.

mentation complexity is between that of a decoder and an encoder.

Given the complexity of the causally optimal algorithm, it is interesting to examine the benefit of error accumulation tracking. This can be evaluated by applying the algorithm of Section 3 to the mixed-mode case, since the only difference is the accumulated error term a_i. Surprisingly, the results of this *memoryless* mixed-mode partitioning algorithm are almost identical. Figure 5 shows the relevant PSNR values for the "Mobile" sequence; the difference is in general less than 0.1 dB and the curves can hardly be distinguished. It turns out that this holds for a wide range of bit rates (e.g. down to 50%) and slice sizes, although the difference increases slightly to 0.2-0.3 dB.

This is a very important result, as it implies that we can dispense completely with the error accumulation calculation and its associated computational complexity, for a minimal cost in performance: the quality degradation between the causally optimal and memoryless algorithms will be perceptually insignificant, across the spectrum of slice sizes and partition rates. This property is hinted by Figure 4 upon closer examination: the upper and lower curves are almost identical, except from a vertical shift. In order for the accumulated error to affect the partitioning decisions, either the slope of the R(D) curves or the overall accumulated error distribution across a picture would have to be significantly affected; this, however, is not the case.

Finally, we examine the performance of the slice-based optimization algorithm discussed in Section 3 in a mixed-mode coding environment. Since, as was mentioned before, slice-based optimization is a purely rate-based operation and does not take into account the distortion, there is no difference whether or not the accumulated error is tracked. Figure 5 depicts the results obtained on the "Mobile" sequence, with the same coding and partitioning parameters as before. We see that the slice-based algorithm is inferior by about 1 dB. The complexity, however, is significantly reduced as well, as the Lagrangian optimization iteration is avoided.

5. CONCLUDING REMARKS

The problem of optimal data partitioning of MPEG-2 coded video was analyzed in an operational rate-distortion context. An optimal algorithm based on Lagrangian optimization was derived for intra-only coding, while a faster but suboptimal algorithm using slice-based optimization was shown to perform quite close (within 0.6 dB on the average).

For the mixed-mode case (I, P, and B pictures) the optimal algorithm was characterized and shown to possess significantly high complexity and delay, as a complete group of pictures was required to be processed at a time. As an alternative, a causally optimal algorithm was proposed, in which only the accumulated error from past pictures was taken into account (with the corresponding error propagated to future pictures ignored). This algorithm is optimal for intra-coded pictures. The simulation results showed that the algorithm performs quite well (comparable to intra-only partitioning), with P and B pictures having about 1 dB higher quality than I ones.

It was then shown that tracking the error accumulation from one frame to the next does not actually benefit the partitioning process in any significant way, and hence that a memoryless algorithm employing Lagrangian optimization is sufficient. This is an important result as it drastically reduces the complexity of the algorithm, and seems to hold across the range of partition rates and slice sizes. Further theoretical investigation is required to verify the general validity of this property. Finally, slice-based optimization on mixed-mode coding was shown to perform within 1 dB of the causally optimal algorithm, and hence represents a reasonably good tradeoff between performance and complexity.

6. REFERENCES

[1] Generic Coding of Moving Pictures and Associated Audio (MPEG-2). ITU-T Draft Recommendation H.262, ISO/IEC 13818-2 Draft International Standard, 1994.

[2] K. Ramchandran and M. Vetterli. Rate-distortion optimal fast thresholding with complete JPEG/MPEG decoder compatibility. In *Proceedings of the Picture Coding Symposium '93*, March 1993.

[3] K. Ramchandran and M. Vetterli. Rate-distortion optimal fast thresholding with complete JPEG/MPEG decoder compatibility. *IEEE Transactions on Image Processing* (Special Issue on Video Sequence Compression), 1994. To appear.

[4] Didier LeGall. MPEG: A Video Compression Standard for Multimedia Applications. *Communications of the ACM*, 34(4):46–58, April 1991.

[5] Y. Shoham and A. Gersho. Efficient bit allocation for an arbitrary set of quantizers. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(9):1445–1453, 1988.